

## CLAIMS

1. (Currently Amended) One or more processor-accessible tangible storage media comprising processor-executable instructions stored thereon that, when executed, direct a device to perform a method comprising:

converting a plurality of programs into corresponding intermediate language programs, the plurality of programs being written in different programming languages but according to a common language specification;

providing exposing resources of an operating system or an object model service via functions of a programming interface layer to the plurality of intermediate language  
~~for developing~~ programs, the functions being organized into groups according to type,  
the groups programming interface having:

~~multiple groups of types and including:~~

a first group of types related to core file system concepts;

a second group of types related to entities that a human being can  
contact;

a third group of types related to documents;

a fourth group of types common to multiple kinds of media;

a fifth group of types specific to audio media;

a sixth group of types specific to video media;

a seventh group of types specific to image media;

an eighth group of types specific to electronic mail messages; and

a ninth group of types related to identifying particular locations, ~~wherein~~  
~~the programming interface provides callable multiple functions, wherein each one of the~~

~~multiple groups of types provides a corresponding set of related ones of the multiple functions, one of the types being configured to communicate in first a form compatible with a first interface and one of the other types being configured to communicate in a second form compatible with a second interface, the first and second forms being incompatible; and~~

~~a component configured to convert communications in the first form compatible with the first interface to the second form compatible with the second interface which is incompatible with the first form~~

wherein the program interface layer resides on top of a common language run time layer that hands calls to the functions of the program interface layer by the plurality of intermediate language programs to the operating system or the object model service for execution.

2. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises:~~ a tenth group of types related to moving data between file systems.

3. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises:~~ a tenth group of types related to creating and managing rules for generating notifications.

4. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises:~~ a tenth group of types describing types defined in all the other groups of types.

5. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types related to base types that form a foundation to support all the other groups of types.

6. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types common to multiple kinds of messages, including the electronic mail messages; and an eleventh group of types specific to facsimile messages.

7. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types related to annotations; and an eleventh group of types related to notes;

8. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types related to installed programs; and an eleventh group of types related to installed games.

9. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types related to actions taken by a user; and an eleventh group of types related to maintaining and accessing help information.

10. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types related to a natural language search engine.

11. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types related to tasks in a user interface to let a user know what actions the user can perform when navigating the user interface.

12. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types related to user tasks.

13. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types related to services that can be accessed.

14. (Previously Presented) A tangible storage media as recited in claim 13, wherein the services can be accessed over a network.

15. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types related to identifying access rights.

16. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types related to calendar types.

17. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types related to creating and managing event monitoring and resultant actions.

18. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~: a tenth group of types used for interop for each of the first through ninth groups of types.

19. (Currently Amended) A tangible storage media as recited in claim 1, wherein the group further include ~~programming interface further comprises~~: an additional group of types for each of the first through ninth groups of bytes, wherein each of the additional groups of types are for interop.

20. (Currently Amended) A tangible storage media as recited in claim 1, wherein the group further include ~~programming interface further comprises~~: a tenth group of types related to files stored in a file system.

21. (Currently Amended) A tangible storage media as recited in claim 1, wherein the groups further include ~~programming interface further comprises~~; a tenth group of types related to a category hierarchy.

22. (Currently Amended) A system framework comprising:  
an operating system having a plurality of resources;  
a common language runtime layer running on top of the operating system that enables at least one intermediate language application to access at least one resource of the operating system;

~~means for exposing via an application programming interface layer running on top of the common language runtime layer, for developing programs the application program interface layer including functions that enable the at least one intermediate language application to access the at least one resources of the operating system via the common language runtime layer, the functions including:~~

a first set of functions that represent core concepts of a file system of the system;

~~means for exposing via the programming interface~~ a second set of functions that enable maintaining information regarding entities that can be contacted; and

~~means for exposing via the programming interface~~ a third set of functions that allow document types to be accessed; and

a common language specification that enables a plurality of applications to be translated into intermediate language applications, the plurality of application including at least two applications that are written in different programming languages

~~means for converting communications in a first form compatible with a first interface to a second form compatible with a second interface, the first and second forms being incompatible, some of the functions communicating in the first form and some of the other functions communicating in the second form, wherein at least a portion of the means are implemented in hardware.~~

23. (Currently Amended) A system framework as recited in claim 22, wherein the functions ~~further comprising means for exposing via the programming interface~~ comprises a fourth set of functions related to base types for a plurality of kinds of media; means for exposing via the programming interface a fifth set of functions related specifically to audio media; and means for exposing via the programming interface a sixth set of functions related specifically to video media.

24. (Currently Amended) A system framework as recited in claim 23, wherein the functions ~~further comprising: means for exposing via the programming interface~~ comprises a seventh set of functions related specifically to image media.

25. (Currently Amended) A system framework as recited in claim 22, wherein the functions ~~further comprising means for exposing via the programming interface~~ comprises a fourth set of functions related specifically to electronic mail messages.

26. (Currently Amended) A system framework as recited in claim 22, wherein the functions ~~further comprising means for exposing via the programming interface~~ comprises a fourth set of functions that enable maintaining physical location information.

27. (Currently Amended) A method of organizing a set of application program interface (API) functions ~~types for a file system~~ in a program development computer system into a hierarchical namespace, ~~the file system being one of multiple groups of types included in a programming interface for developing programs,~~ the method comprising:

creating a plurality of groups for the set of API functions according to type ~~from the set of types using the program development computer system,~~ each group containing ~~at least one type that exposes~~ logically related API functions ~~functionality of the application programming interface;~~

assigning a name to each group ~~in the plurality~~ using the program development computer system, wherein one of the groups ~~in the plurality~~ includes at least one API function ~~functionality~~ related to core concepts of the file system, wherein another of the groups ~~in the plurality~~ includes at least one API function ~~functionality~~ related to entities that a human being can contact, wherein another of the groups ~~in the plurality~~ includes at least one API function ~~functionality~~ related to document types that can be stored in the file system, and wherein another of the groups ~~in the plurality~~ includes at least one API function ~~functionality~~ related to multiple kinds of media;

selecting a top level identifier and prefixing the name of each group with the top level identifier using the program development computer system so that the types in each group are referenced by a hierarchical name that includes the selected top level identifier prefixed to the name of the group containing the type ~~wherein a first type or a first function communicates in a first form compatible with a first interface and a second~~



~~type or a second function communicates in a second form compatible with a second interface, wherein the first form and the second forms are incompatible; and~~

~~converting a communication associated with the first type or the first function from the first form to the second form whereby a plurality of diverse applications which use incompatible interfaces can be developed using the programming interface.~~

28. (Currently Amended) A method as recited in claim 27, wherein another of the groups ~~in the plurality~~ includes at least one API function ~~functionality~~ particularly for audio media, wherein another of the groups ~~in the plurality~~ includes at least one function ~~functionality~~ particularly for video media, and wherein another of the groups in the plurality includes at least one API function ~~functionality~~ particularly for image media.

29. (Currently Amended) A method as recited in claim 27, wherein another of the groups in the plurality includes at least one API function ~~functionality~~ related to electronic mail.

30. (Currently Amended) A method as recited in claim 27, wherein another of the groups ~~in the plurality~~ includes at least one API function ~~functionality~~ related to maintaining physical location information.

31. (Currently Amended) A method as recited in claim 27, wherein the assigning comprises: assigning a name of Core to the group that includes at least one API function ~~functionality~~ related to core concepts of the file system so that the hierarchical name for the group that includes at least one API function ~~functionality~~

related to core concepts of the file system is System.Storage.Core; assigning a name of Contacts to the group that includes at least one API function~~-functionality~~ related to entities that a human being can contact so that the hierarchical name for the group that includes at least one API function~~-functionality~~ related to entities that a human being can contact is System.Storage.Contacts; assigning a name of Documents to the group that includes at least one API function~~-functionality~~ related to document types that can be stored in the file system so that the hierarchical name for the group that includes at least one API function~~-functionality~~ related to document types that can be stored in the file system is System.Storage.Documents; and assigning a name of Media to the group that includes at least one API function~~-functionality~~ related to multiple kinds of media so that the hierarchical name for the group that includes at least one API function~~-functionality~~ related to multiple kinds of media is System.Storage.Media.

32. (Currently Amended) A method as recited in claim 27, wherein the assigning comprises: assigning a name of Core to the group that includes at least one API function~~-functionality~~ related to core concepts of the file system so that the hierarchical name for the group that includes at least one API function~~-functionality~~ related to core concepts of the file system is System.Storage.Core; assigning a name of Contact to the group that includes at least one API function~~-functionality~~ related to entities that a human being can contact so that the hierarchical name for the group that includes at least one API function~~-functionality~~ related to entities that a human being can contact is System.Storage.Contact; assigning a name of Document to the group that includes at least one API function~~-functionality~~ related to document types that can be stored in the file system so that the hierarchical name for the group that includes at least

~~one API function-functionality~~ related to document types that can be stored in the file system is System.Storage.Document; and assigning a name of Media to the group that includes at least one API function ~~functionality~~ related to multiple kinds of media so that the hierarchical name for the group that includes at least one API function ~~functionality~~ related to multiple kinds of media is System.Storage.Media.

33. (Currently Amended) A method for organizing a file system in a program development computer system, the method comprising:

creating a first namespace ~~with~~ that includes application program interface (API) functions of an API layer that enable identification of particular physical locations using the program development computer system, the API interface layer running on top of a common language runtime layer to receive API function calls from an intermediate language program; and

creating a second namespace ~~with~~ that includes API functions of the API layer that enable identification of entities that can be contacted by a human being using the program development computer system, wherein the first namespace and the second namespace are included in the file system, the file system being included in a programming interface, ~~one of the functions communicating in a first form and another function communicating in a second form which is incompatible with the first form; and~~

~~converting a communication associated with one of the functions and in the first form to the second form.~~

34. (Currently Amended) A method as recited in claim 33, further comprising: creating a third namespace using the program development computer system-~~with~~ that includes API functions of the API layer that enable documents to be described.

35. (Currently Amended) A method as recited in claim 33, further comprising: creating a third namespace using the program development computer system-~~with~~ that includes API functions of the API layer specific to electronic mail messages.

36. (Currently Amended) A method as recited in claim 33, further comprising: creating a third namespace using the program development computer system-~~with~~ that includes API functions of the API layer common to multiple kinds of media; creating a fourth namespace using the program development computer system-~~with~~ that includes API functions of the API layer specific to audio media; creating a fifth namespace using the program development computer system-~~with~~ that includes API functions of the API layer specific to video media; and creating a sixth namespace using the program development computer system-~~with~~ that includes API functions of the API layer specific to image media.

37. (Currently Amended) A method as recited in claim 33, further comprising: creating a third namespace using the program development computer system-~~with~~ that includes API functions of the API layer that are expected to be used by all other namespaces.

38. (Currently Amended) One or more tangible computer readable media having stored thereon a plurality of instructions that, when executed by a processor, cause the processor to:

create a first namespace ~~with~~ that includes application program interface (API) functions of the API layer that enable identification of particular physical locations, the API interface layer running on top of a common language runtime layer to receive API function calls from an intermediate language program; and

create a second namespace ~~with~~ that includes API functions of the API layer that are expected to be used by the first namespace and a plurality of additional namespaces, wherein the first namespace, the second namespace, and the plurality of additional namespaces are defined to organize a file system, the file system being included in a programming interface for developing programs, one of the that includes API functions of the API layer communicating in a first form and another function communicating in a second form which is incompatible with the first form; and

convert a communication associated with one of the that includes API functions of the API layer and in the first form to the second form.

39. (Currently Amended) One or more tangible computer readable media as recited in claim 38, wherein the instructions further cause the processor to: create a third namespace ~~with~~ that includes API functions of the API layer that enable documents to be described; create a fourth namespace ~~with~~ that includes API functions of the API layer that enable identification of entities that can be contacted by a human being; and create a fifth namespace with that includes API functions of the API layer common to multiple kinds of media.

40. (Currently Amended) One or more tangible computer readable media as recited in claim 39, wherein the instructions further cause the processor to: create a sixth namespace with that includes API functions of the API layer specific to audio media; create a seventh namespace with that includes API functions of the API layer specific to video media; and create an eighth namespace with that includes API functions of the API layer specific to image media.

41. (Currently Amended) One or more tangible computer readable media as recited in claim 38, wherein the instructions further cause the processor to: create a third namespace with that includes API functions of the API layer common to multiple kinds of media; create a fourth namespace with that includes API functions of the API layer specific to audio media; create a fifth namespace with that includes API functions of the API layer specific to video media; and create a sixth namespace with that includes API functions of the API layer specific to image media.

42. (Currently Amended) A method comprising:

calling one or more first application program interface (API) functions of an API layer that is running on top of a common language runtime layer ~~using a program development computer system,~~ the one or more first API functions that enable documents to be described;

calling one or more second API functions of the API layer ~~using the program development computer system~~ that are core functions expected to be used by the one or more first functions as well as a plurality of additional functions, wherein the one or

more first functions, the one or more second functions, and the plurality of additional functions ~~are defined to organize in~~ a file system in the program development computer system, the file system being included in the API ~~a programming interface, one of the functions communicating in a first form and another function communicating in a second form which is incompatible with the first form; and~~

~~converting a communication associated with one of the functions and in the first form to the second form.~~

43. (Currently Amended) A method as recited in claim 42, further comprising: calling one or more third API functions of the API layer common to multiple kinds of media using the program development computer system.

44. (Currently Amended) A method as recited in claim 43, further comprising: calling one or more fourth API functions of the API layer specific to audio media using the program development computer system; calling one or more fifth API functions of the API layer specific to video media using the program development computer system; and calling one or more sixth API functions of the API layer specific to image media using the program development computer system.

45. (Currently Amended) A method as recited in claim 42, further comprising: calling one or more third API functions of the API layer using the program development computer system that enable identification of entities that can be contacted by a human being; and calling one or more fourth API functions of the API layer using the program development computer system that enable identification of particular physical locations.

46. (Currently Amended) A method as recited in claim 42, further comprising: calling one or more third API functions of the API layer specific to electronic mail messages using the program development computer system.

47. (Currently Amended) A method, comprising:  
receiving one or more calls to one or more first application program interface (API) functions of the API layer that is running on top of a common language runtime layer using a program development computer system that, the one or more first API functions enable identification of entities that can be contacted by a human being; and  
receiving one or more calls to one or more second API functions of the API layer that are core functions expected to be used by the one or more first API functions of the API layer as well as a plurality of additional API functions of the API layer using the program development computer system, wherein the one or more first API functions of the API layer, the one or more second API functions of the API layer, and the plurality of additional API functions of the API layer are defined to organize a file system in the program development computer system, the file system being included in a programming interface, ~~one of the functions communicating in a first form and another function communicating in a second form which is incompatible with the first form; and~~  
~~converting a communication associated with one of the functions and in the first form to the second form.~~

48. (Currently Amended) A method as recited in claim 47, further comprising: receiving one or more calls using the program development computer system to one or



more third API functions of the API layer that enable documents to be described; receiving one or more calls using the program development computer system to one or more fourth API functions of the API layer common to multiple kinds of media; and receiving one or more calls using the program development computer system to one or more fifth API functions of the API layer that enable identification of particular physical locations.

49. (Currently Amended) A method as recited in claim 48, further comprising: receiving one or more calls using the program development computer system to one or more sixth API functions of the API layer specific to audio media; receiving one or more calls using the program development computer system to one or more seventh API functions of the API layer specific to video media; receiving one or more calls using the program development computer system to one or more eighth API functions of the API layer specific to image media and receiving one or more calls using the program development computer system to one or more ninth API functions of the API layer specific to electronic mail messages.

50. (Currently Amended) One or more tangible computer readable media having stored thereon a plurality of instructions that, when executed by a processor, cause the processor to:

receive one or more calls to one or more first application program interface (API) functions of the API layer that enable identification of entities that can be contacted by a human being, the API layer running on top of a common language runtime layer; and

receive one or more calls to one or more second API functions of the API layer common to multiple kinds of media, wherein the one or more first API functions of the API layer and the one or more second API functions of the API layer are defined to organize a file system, the file system being included in a programming interface for programming programs, one of the API functions of the API layer communicating in a first form and another function communicating in a second form which is incompatible with the first form; and

convert a communication associated with one of the API functions of the API layer and in the first form to the second form.

51. (Currently Amended) One or more tangible computer readable media as recited in claim 50, wherein the instructions further cause the processor to: receive one or more calls to one or more third API functions of the API layer that are core API functions of the API layer expected to be used by the one or more first API functions of the API layer, the one or more second API functions of the API layer, and a plurality of additional API functions of the API layer.

52. (Currently Amended) One or more tangible computer readable media as recited in claim 50, wherein the instructions further cause the processor to: receive one or more calls to one or more third API functions of the API layer that enable identification of particular physical locations; receive one or more calls to one or more fourth API functions of the API layer that enable documents to be described; and receive one or more calls to one or more fifth API functions of the API layer specific to electronic mail messages.

53. (Currently Amended) One or more tangible computer readable media as recited in claim 50, wherein the instructions further cause the processor to: receive one or more calls to one or more third API functions of the API layer specific to audio media; receive one or more calls to one or more fourth API functions of the API layer specific to video media; and receive one or more calls to one or more fifth API functions of the API layer specific to image media.